Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Computers & Operations Research 32 (2005) 2255–2269
computers & operations research
www.elsevier.com/locate/dsw

Enhancing router QoS through job scheduling with weighted shortest processing time-adjusted

Nong Ye*, Zhibin Yang, Ying-Cheng Lai, Toni Farley

Information and Systems Assurance Laboratory, Arizona State University, P.O. Box 875906, Tempe, AZ 85287-5906, USA

Abstract

Most routers on the Internet employ a first-in-first-out (FIFO) scheduling rule to determine the order of serving data packets. This scheduling rule does not provide quality of service (QoS) with regards to the differentiation of services for data packets with different service priorities and the enhancement of routing performance. We develop a scheduling rule called Weighted Shortest Processing Time–Adjusted (WSPT-A), which is derived from WSPT (a scheduling rule for production planning in the manufacturing domain), to enhance router QoS. We implement a QoS router model based on WSPT-A and run simulations to measure and compare the routing performance of our model with that of router models based on the FIFO and WSPT scheduling rules. The simulation results show superior QoS performance when using the router model with WSPT-A.

Keywords: Quality of service; Router; Scheduling rules; Weighted shortest processing time and delay stability

1. Introduction

Routers in the Internet are necessary to support networking and data communication. A router receives data packets from sources on the Internet at its input port(s) and sends them out to requested destinations through its output port(s). Because a router's data transmission rate is limited by available bandwidth, it typically uses a buffer or queue of limited storage capacity to keep incoming packets awaiting service. Packets are removed from the queue for servicing when bandwidth becomes available. If a data packet arrives at a router when its queue is full, the packet is dropped. Most routers on the Internet today use the first-in-first-out (FIFO) scheduling rule to determine the

* Corresponding author. Tel.: +1-480-965-7812; fax: +1-480-965-8692.
E-mail address: nongye@asu.edu (N. Ye).

0305-0548/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2004.03.001

order in which queued packets will be served [1]. By this rule, the first data packet to arrive at a router is placed at the front of its queue, and is then first to be removed for servicing.

Quality of service (QoS) requires the differentiation of services for data packets with different priorities and the enhancement of performance metrics, such as delay, packet loss, throughput, and so on [2–8]. Various kinds of network applications run on the Internet, including email, web browsing, teleconferencing, IP telephony, etc. Data packets for these applications have varying characteristics and QoS requirements. For example, some applications place no 'hard' constraints on delay. Others, such as audio and video applications, are time-dependent and place strict constraints on delay and packet loss. QoS on the Internet requires that data packets with different QoS requirements are provided differentiated services based on those requirements. The FIFO scheduling rule provides services to data packets based on their arrival times without consideration of their QoS requirements. Moreover, if we consider the scheduling of services for data packets as a job scheduling problem, theories for job scheduling in the manufacturing domain show that FIFO does not achieve optimization of job scheduling performance with regard to QoS performance metrics [9]. Thus, Internet routers using FIFO scheduling do not provide QoS.

Several QoS architectures for the Internet have been proposed, including Integrated Service (IntServ) and Differentiated Services (DiffServ) [2–8,10]. IntServ provides QoS on a per-flow basis. An end-to-end (source-to-destination) bandwidth reservation is required to firmly guarantee service to an individual data flow. The implementation of IntServ requires the support of a range of complex mechanisms, such as packet scheduling, packet classification, admission control, and path reservation. IntServ is not scalable because of the management overhead required to maintain the state of each flow. Therefore, it is not practical for the Internet, which carries a large number of individual flows.

DiffServ addresses the scalability problem of IntServ by providing QoS on a per-aggregation basis. DiffServ divides the Internet into domains with edge and core routers that perform different functions. The edge routers of a domain classify, police, and mark data packets based on certain administrative policies. Core routers inside the domain provide per-hop QoS corresponding to the type of aggregate traffic. To differentiate services for data packets with different classes of QoS requirements or service priorities in a core router, multiple queues are maintained. Packets are placed into queues corresponding to their class. Each queue services packets using FIFO. In DiffServ, there is no need to reserve bandwidth on an end-to-end connection path.

Both of the proposed architectures have shortcomings with regards to QoS. For example, the per-connection end-to-end bounds on delay for Intserv depend on the number of hops, or routers, on a connection path. However, hop count is uncertain due to routing dynamics. Thus, an end user cannot expect to have a delay bound on an absolute time basis. By reserving bandwidth for an individual flow, Intserv does provide better QoS than without bandwidth reservation, but does not guarantee QoS. DiffServ provides even better QoS for the high-priority class of data packets, but again without guarantees. Thus, neither Intserv nor Diffserv provide absolute guarantees for QoS.

Zhang surveys a number of scheduling rules to provide per-connection end-to-end QoS in packet-switching networks (Internet) [11]. Many scheduling rules and associated performance problems have also been studied in the context of real-time and queuing systems with the goals of optimizing various performance measures such as delay and throughput [9,12,13]. However, existing job scheduling rules target the optimal performance on average, in sum or with an upper bound, for a population of jobs. Scheduling rules that optimize average performance do not necessarily produce performance stability, which also plays an important part in end-users' QoS satisfaction. For example, end users may feel

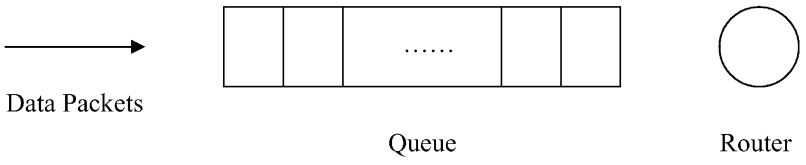

Fig. 1. A generic router model.

more frustrated when delays vary to a large extent at different times than when delays stay at a stable and thus predictable level.

If we consider a performance measure for a population of jobs that follow a certain probability distribution, it is typically desirable to have both an average close to a target value and a small variance. A small variance implies performance stability and thus performance predictability. Thus, for a given performance metric such as delay, a small variance with a slightly off target average can illicit higher end-user satisfaction than an on target average with a high variance. Moreover, the performance stability and predictability of individual service providers, such as routers, enables end users to proactively plan their tasks to meet QoS requirements, rather than being passive and having little control. A measure for performance stability is delay jitter [11]. Some studies focus on the end-to-end delay bounds of traffic regulated, packet-switched networks [14–17]. Still, scheduling rules for bounding or minimizing delay jitter while maintaining a desirable average delay level are not well established.

In this study, we investigate a scheduling rule for routers that aims to provide performance stability for QoS while maintaining a desirable average performance. In the following sections of this paper, we first describe the FIFO and Weighted Shortest Processing Time (WSPT) scheduling rules. Next we introduce our WSPT-Adjusted (WSPT-A) scheduling rule and describe how we derive this rule from WSPT. We present implemented models of routers using these three rules, and describe experiments that test the QoS performance of these models under different traffic conditions. Finally, we present our simulation results and compare the QoS performance of the router models.

2. Router models with FIFO, WSPT, and WSPT-A

We present models of routers using the FIFO and WSPT scheduling rules, followed by the description of a router model using WSPT-A, which is derived from WSPT. These router models are used in our study to compare the QoS performance of the three scheduling rules. For each of our router models, we implement the corresponding scheduling rule in the queue of the generic router shown in Fig. 1.

2.1. Router model with FIFO

We choose the router model with FIFO to compare performance with our model because it is a common router model in practical use on the Internet. This model employs the simple FIFO scheduling rule in the queue, and processes packets in the order in which they are received. In this case, arrival time is the only factor considered when ordering packets for service.

2.2. Router model with WSPT

WSPT is a scheduling rule developed for production planning in the manufacturing domain. For a given set of jobs that are ready for processing on a single machine at time = 0, WSPT minimizes the total weighted completion time [12]. The completion time of a job is the time it takes to process it, which includes waiting and servicing time. Delay is defined as the time a job is completed less its start time. Thus, WSPT minimizes the total weighted delay.

Using WSPT, the priority of a job is given by the ratio of the weight factor to the service time of the job as follows:

$$p_i = \frac{w_i}{t_i}, \tag{1}$$

where p_i is the service priority of job i, w_i its priority weight, and t_i its processing time. Jobs are served in decreasing order of service priority. By incorporating the weight factor into the computation of a job's service priority, WSPT is capable of differentiating services for jobs with different priority weights. By incorporating the processing time of a job, WSPT minimizes the weighted delay for a set of jobs.

WSPT assumes data packets contain information of priority weight, which can be specified in the Type-Of-Service (TOS) field of a TCP/IP header. The service time of a data packet can be obtained as follows:

$$t_i = \frac{S_i}{r}, \tag{2}$$

where S_i denotes the packet size of i in bits, and r denotes the bandwidth (service rate) of the output port. Substituting Eq. (2) into Eq. (1), the priority of a data packet can be computed as follows:

$$p_i = \frac{w_i \times r}{S_i}. \tag{3}$$

As long as there is enough space available in the queue, an incoming data packet is placed into the queue according to its priority. If the queue does not have enough space available, the priority of the incoming data packet is compared with that of the last data packet in the queue. If the incoming packet has a higher priority, the last packet is de-queued and dropped by the router. This comparison and dropping of data packets continues until there is enough space available for the incoming data packet, or the last data packet in the queue has a higher priority.

Since data packets continuously arrive at the router, low-priority packets in the queue may wait for a short or long time, or be dropped before they are served, depending on the number and priorities of arriving packets. The dynamic insertion of packets in the queue introduces instability in the router by creating a large variance in delays.

2.3. Router model with WSPT-A

To overcome instability problems associated with WSPT, we modify it by introducing an additional term to increases the compensation in the priority of a data packet as the waiting time of the data packet increases. The adjusted scheduling rule, called WSPT-Adjusted or WSPT-A, uses the

following formula to compute the priority of a data packet:

$$p_i = \frac{w_i \times r}{S_i} c_i, \tag{4}$$

where c_i is an exponential compensation term for data packet i given by the following:

$$c_i = e^{-\lambda P/(T_i+\eta P)}, \tag{5}$$

where T_i stands for the waiting time of data packet i in the queue, λ and η are constants, and P the average service time of data packets passing through the router. P can be estimated from the ratio of average packet size to output port bandwidth, and used as a constant in Eq. (5).

Using an exponential compensation term, the compensation value falls in the range [0, 1]. Given λ, η and P, the value of the compensation term depends on waiting time. The compensation term increases to 1 as waiting time increases from 0 to infinity. We let the exponential compensation term change between some initial level α and its maximum value of 1. The initial level is the compensation value when the waiting time is zero as follows:

$$\alpha = e^{-\lambda P/(0+\eta P)} \tag{6}$$

that is, T_i equals 0. The parameters λ and η can be determined by setting the level of compensation for a waiting time that is considered to reach the limit of tolerance. We can express this waiting time in terms of n times the average service time P and set the corresponding compensation value to β. Thus, when T_i increases from 0 to nP, the compensation value increases from α to β as follows:

$$\beta = e^{-\lambda P/(nP+\gamma P)}. \tag{7}$$

By solving Eqs. (6) and (7) for λ and η, we get

$$\lambda = -\frac{\ln\alpha \ln\beta}{\ln\alpha - \ln\beta} n, \tag{8}$$

$$\eta = \frac{\ln\beta}{\ln\alpha - \ln\beta} n. \tag{9}$$

Formulas (8) and (9) are used to obtain λ and η using α, β and n. The values of λ and γ are then used to compute the priority of a data packet. Priorities are recomputed with updated waiting time values whenever an incoming packet needs to be queued. After recomputing the priorities of packets in the queue, they are resorted to maintain a decreasing order of priority.

3. Simulation and experiments

To examine QoS performance, we implement the three router models using OPNET Modeler software [13]. We then conduct simulation experiments under various traffic conditions. In this section, we first describe the implementation of the three router models, followed by a description of our simulation experiments.

3.1. Implementation of router models

The router model implementation is shown in Fig. 2. Each model consists of two input ports, an IP forwarder module, an output port, and a packet sink labeled “egress”. Each input port is associated

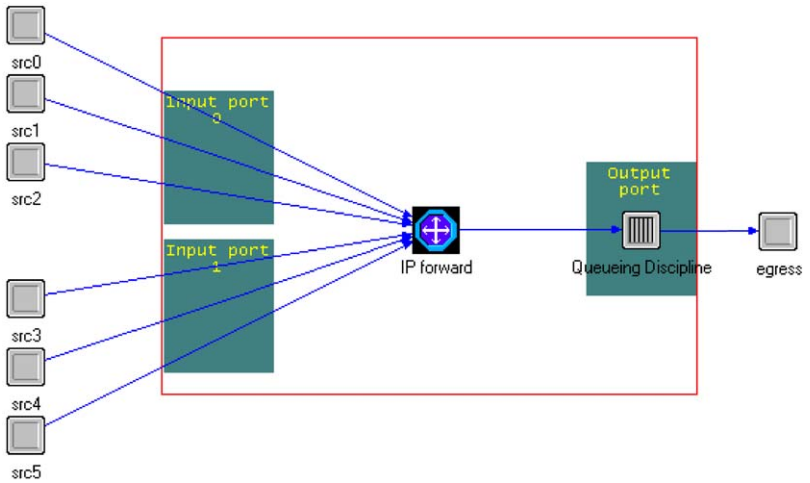

Fig. 2. Implementation of the router models using OPNET Modeler.

Table 1
Implementation configuration of the three router models

Router model	Service rate (b/s)	Queue capacity (b)	Priority weight		α	nP	β
			High	Low			
FIFO	640,000	550,000	N/A	N/A	N/A	N/A	N/A
WSPT	640,000	550,000	5	2	N/A	N/A	N/A
WSPT-A	640,000	550,000	5	2	0.3875	60P	0.95

with three traffic sources, for a total of six (src0 to src5). Each traffic source generates a stream of data packets with a high- or low-priority weight. The priority of a data packet is marked in the ToS field of the IP header. We set ToS to 7 for high-priority, and 0 for low-priority. In general, a Poisson process is a good characterization of a random arrival process. Thus, we use a Poisson process to generate data packets at each traffic source, so the inter-arrival time of packets follows an exponential distribution. A normal distribution is used to determine the packet sizes. The mean data arrival rate (bits/second) for each traffic source can be determined by the ratio of the mean packet size (bits) to the mean inter-arrival time (seconds).

The IP forwarder module forwards packets from the inputs to the output port, where a queueing discipline module is implemented with a queue and a scheduling rule. The different scheduling rules (FIFO, WSPT, and WSPT-A) are implemented for different router models. Finally, the packet sink collects output packets.

Table 1 shows the identical configurations of the router models, including the service rate in bits/second (b/s), queue capacity in bits (b), and the weight values of high- and low-priority data packets. For WSPT and WSPT-A, priority weights of 5 and 2 are assigned to high- and low-priority

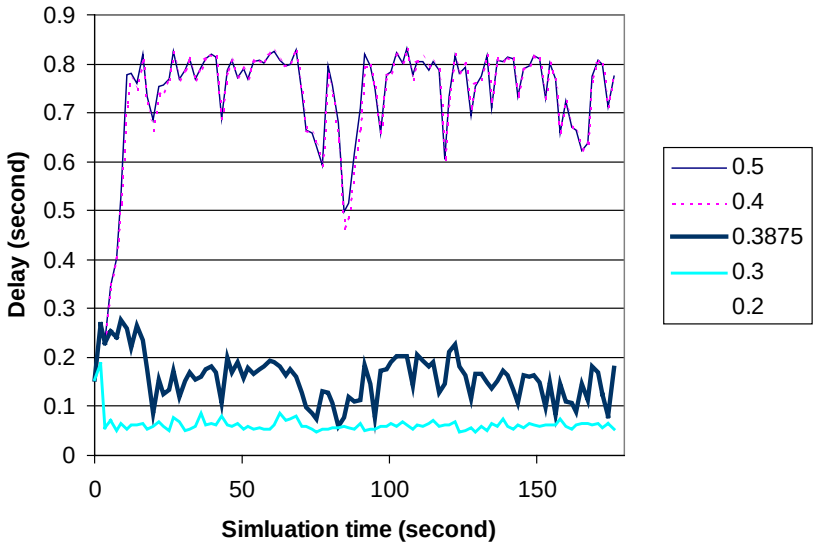

Fig. 3. Delay performance of WSPT-A router from preliminary simulations.

packets identified by ToS values. Thus, 2.5 is the ratio of high- to low-priority weight, which gives a higher service priority to more high than low-priority weight data packets.

For WSPT-A, we set β to 0.95 for a waiting time of 60 times the average service time ($60P$), since the queue capacity is about 55 times the average packet size, a waiting time of 60P is approximately equivalent to the worst case waiting time for FIFO. To select an appropriate compensation level α, we run a series of preliminary simulations under a heavy traffic condition (described in the next section). For each simulation, the service rate, queue capacity and weight values are set to the levels shown in Table 1. The values of α vary from 0.2 to 0.5. Figs. 3 and 4 show the delay time and packet loss for α values of 0.2, 0.3, 0.3875, 0.4 and 0.5. When α is set to 0.3875, the router model with WSPT-A demonstrates an effective control of packet delay with a low level of packet loss. Hence, we set α to 0.3875.

3.2. Simulation experiments

Each model is tested under two types of traffic conditions: heavy and light. In the heavy traffic condition, the mean data arrival rate of high-priority data packets exceeds the service rate of the router so that low-priority packets have little chance of being served. In the light traffic condition, this arrival rate is lower than the service rate so that low-priority packets have a good chance of being served. For each condition, each traffic source generates packets using the same normal distribution for packet size with a mean of 10,000 bits and variance of 2000 bits. Using the mean size of data packets and the service rate of the router in Table 1, we obtain an average service time P of 0.0156 s for all router models. Sources 0, 1, 3, and 4 generate high-priority packets, and 2 and 5 generate low-priority packets.

In the heavy traffic condition shown in Table 2, each input port generates high-priority data packets at an average arrival rate of 350,000 b/s (250,000 b/s + 100,000 b/s from two traffic sources) and

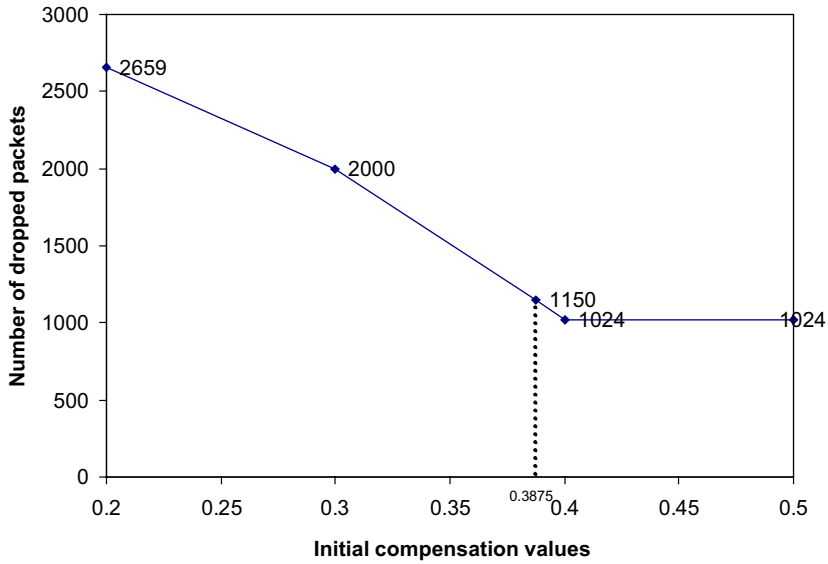

Fig. 4. Packet loss of WSPT-A router from preliminary simulations.

Table 2
Characteristics of data packets in the heavy traffic condition

Traffic source	Priority	Interface	Inter-arrival time		Mean arrival rate (b/s)
			Distribution	Mean (s)	
0	High	0	Exponential	0.04000	250,000
1	High	0	Exponential	0.10000	100,000
2	Low	0	Exponential	0.06667	150,000
3	High	1	Exponential	0.04000	250,000
4	High	1	Exponential	0.10000	100,000
5	Low	1	Exponential	0.06667	150,000

low-priority at the rate of 150,000 b/s. In total, high-priority data packets are generated at the average rate of 700,000 b/s, which is higher than the service rate of the router.

In the light traffic condition shown in Table 3, each input port generates high-priority packets at 150,000 b/s ($75,000 \times 2$ b/s from two traffic sources) and low-priority at 150,000 b/s. The total high-priority packets are generated at an average rate of 300,000 b/s, which is lower than the service rate of the router.

We run one simulation run for each of the three router models in each of the two traffic conditions, totaling six simulations. Each simulation runs for 180 s.

Table 3
Characteristics of data packets in the light traffic condition

Traffic source	Priority	Interface	Inter-arrival time		Mean arrival rate (b/s)
			Distribution	Mean (s)	
0	High	0	Exponential	0.13333	75,000
1	High	0	Exponential	0.13333	75,000
2	Low	0	Exponential	0.06667	150,000
3	High	1	Exponential	0.13333	75,000
4	High	1	Exponential	0.13333	75,000
5	Low	1	Exponential	0.06667	150,000

3.3. Measures of QoS performance

QoS has three main attributes: timeliness, precision, and accuracy [18,19]. If we consider a service request as a process, for a given input, timeliness measures how fast an output is produced and precision measures how much output is given. The router models in this study address the timeliness and precision attributes of QoS. Measures of timeliness include delay, delay jitter, and response time. Measures of precision include throughput, bandwidth, and packet loss (drop) rate.

We collect packet delay (in seconds) to measure timeliness. To collect this performance data, we use a data collection mode in OPNET Modeler, called the bucket mode, where the total simulation time is evenly divided into time intervals. The bucket contains raw data collected during a time interval, and gives the average of this data as a measurement. In our 180 s simulations, we collect 100 values of delay with 1.8 s time intervals. We collect this measure for high and low priority packets separately.

We collect the percentage packet loss rate and throughput to measure precision. Packet loss rate measures packets that are rejected or dropped at the queue. An incoming data packet can be rejected if there is not enough available space in the queue, or a data packet already in the queue can be dropped to make room for a higher priority packet. The packet loss rate is a ratio of rejected or dropped packets to total packets received at a given time. Throughput is defined as the data rate (b/s) from the router's output port. Throughput is also collected using the bucket mode. Both loss rate and throughput are collected for high and low-priority packets separately.

4. Results and discussions

In this section, we present simulation results for the heavy and light traffic conditions, followed by a discussion of these results.

4.1. Results for heavy traffic condition

The delay performance of high-priority data packets for all three router models is shown in Fig. 5. We see that the router model with WSPT-A produces the lowest mean delay, with a much smaller

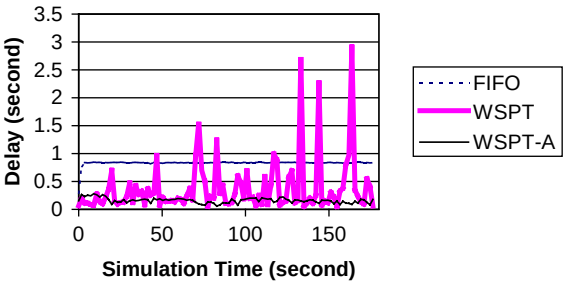

Fig. 5. Delay of high-priority data packets in the heavy traffic condition.

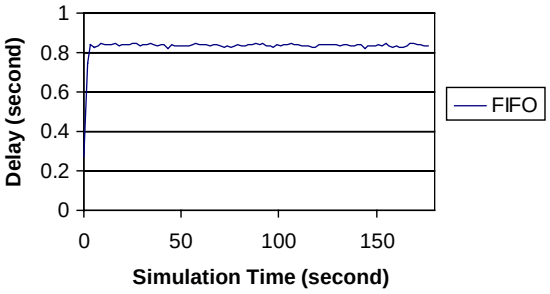

Fig. 6. Delay of low-priority data packets using FIFO in the heavy traffic condition.

variance (and thus, greater stability) than with WSPT. Fig. 5 clearly demonstrates an instability problem using WSPT. While FIFO also has a small variance in delay, its mean delay is much higher than WSPT-A. Hence, for the heavy traffic condition, the router model with WSPT-A demonstrates the best performance for delay in high-priority data packets.

The delay performance of low-priority data packets from the router model with FIFO is shown in Fig. 6. WSPT serves no low-priority packets, and WSPT-A serves very few, thus their delay performance is not measurable in comparison with the FIFO model. The router model with WSPT-A serves a few low-priority data packets because their priorities increase as their waiting times increases if they are not pushed out of the queue by higher priority packets. The served low-priority data packets experience extremely large delays.

The packet loss rates of high- and low-priority data packets, and all packets combined, at the end of the simulation are summarized for all router models in Table 4. For high-priority data packets, the router models with WSPT and WSPT-A produce much lower packet loss rates than that of FIFO. The loss rate with WSPT-A is slightly higher than with WSPT because of the few low-priority data packets that get served under WSPT-A. The trade-off for serving more high-priority packets with WSPT-A and WSPT is shown in their higher loss rates of low-priority packets. For all data packets combined, the three router models produce almost the same packet loss rate because they are all subject to the limited service capacity of the router.

Similar conclusions can be drawn from the throughput performance of the three router models shown in Figs. 7, 8 and 9. The throughput performance of WSPT and WSPT-A is much higher

Table 4
Packet loss rate for all router models in the heavy traffic condition

	FIFO	WSPT	WSPT-A
High priority data packets			
Lost packets	4555	1043	1150
Arrived packets	12,583	12,583	12,583
Packet loss rate	36.2%	8.3%	9.1%
Low priority data packets			
Lost packets	1945	5483	5353
Arrived packets	5488	5488	5488
Packet loss rate	35.4%	99.9%	97.5%
All data packets			
Lost packets	6500	6526	6503
Arrived packets	18,071	18,071	18,071
Packet loss rate	36.0%	36.1%	36.0%

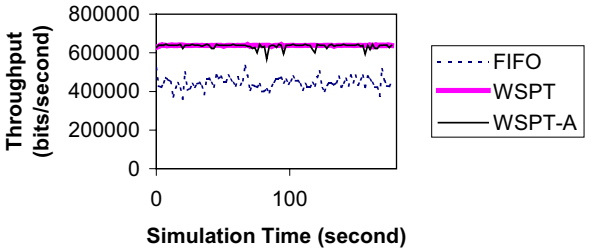

Fig. 7. Throughput of high-priority data packets in the heavy traffic condition.

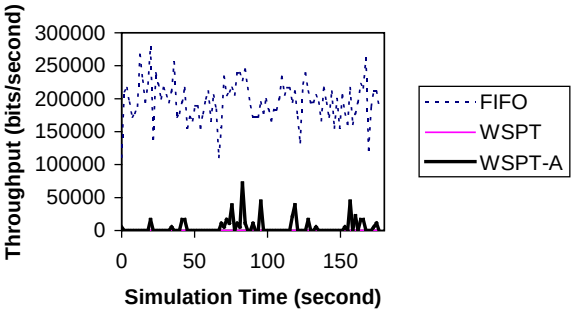

Fig. 8. Throughput of low-priority data packets in the heavy traffic condition.

for high-priority data packets, and lower for low-priority packets when compared with FIFO. The throughput performance of all data packets is about the same for all three models (640,000 b/s) because they are all subject to the limited service capacity of the router.

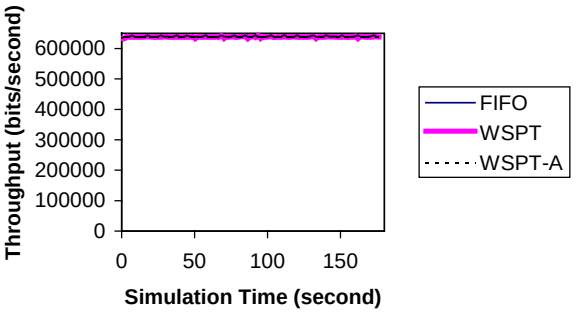

Fig. 9. Throughput of all data packets in the heavy traffic condition.

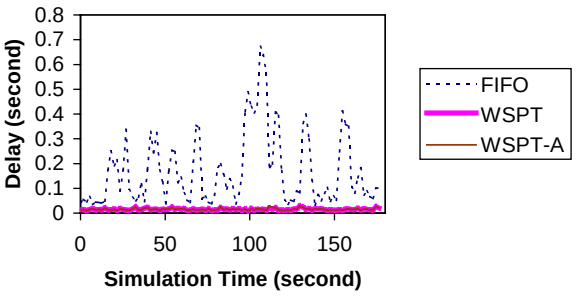

Fig. 10. Delay of high-priority data packets in the light traffic condition.

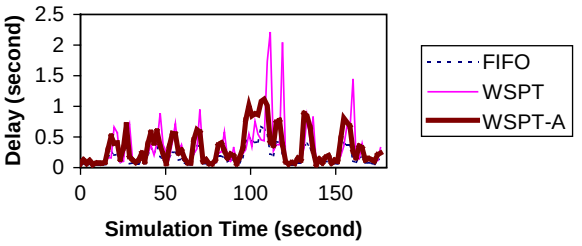

Fig. 11. Delay of low-priority data packets in the light traffic condition.

4.2. Results for light traffic condition

The delay performance of high- and low-priority data packets for all three router models is shown in Figs. 10 and 11. In Fig. 10, WSPT overlaps with WSPT-A, indicating a similar performance for high-priority packets, which is much better than that of FIFO. Conversely, the delay performance for low-priority is slightly better with FIFO as shown in Fig. 11. WSPT and WSPT-A produce comparable delay performance for low-priority data packets. Again, the router model with WSPT produces a larger variance in delay than the router model with WSPT-A.

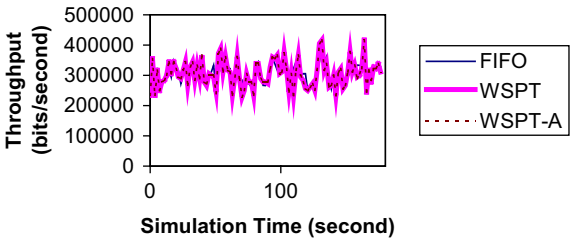

Fig. 12. Throughput of high-priority data packets in the light traffic condition.

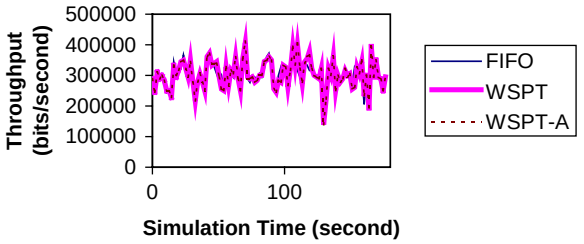

Fig. 13. Throughput of low-priority data packets in the light traffic condition.

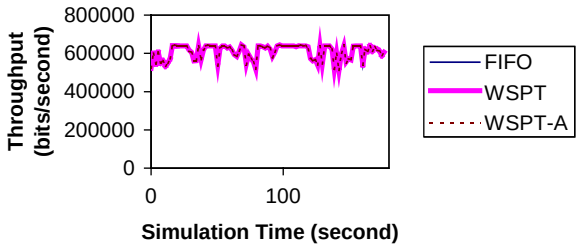

Fig. 14. Throughput of all data packets in the light traffic condition.

In the light traffic condition, there is no packet loss from the three router models because the data arrival rate of all packets is smaller than the service and queue capacity of the router.

Throughput performance for all router models is shown in Figs. 12, 13, and 14. The models demonstrate similar throughput performance in all cases because both high- and low-priority packets get the bandwidth they need in the light traffic condition.

4.3. Discussion of results

The benefits of using WSPT-A over FIFO and WSPT are evident in the heavy traffic condition. Overall, the router models with WSPT and WSPT-A provide much better QoS performance in terms of service differentiation than with FIFO. From Fig. 5, we observe that by introducing the exponential compensation term in WSPT-A, we overcome the instability problem with WSPT, and thus provide performance stability. The cost of adding this term is a slightly higher loss rate of high-priority

packets in heavy traffic as shown in Table 4. With regard to packet loss rate and throughput, WSPT and WSPT-A are comparable.

For the light traffic condition, the router models with WSPT-A and WSPT provide better QoS for high-priority data packets than with FIFO, in terms of delay. This is directly related to the ability of WSPT and WSPT-A to differentiate services between high- and low-priority data packets. Again, WSPT-A overcomes the instability problem associated with WSPT and produces a low level of delay with a small variance. Because the router is able to handle all incoming traffic in the light traffic condition, there is no difference in performance between the router models with regards to packet loss and throughput.

5. Conclusion

Although the router model with FIFO is commonly used on the Internet today, we demonstrated in this study that by using the WSPT-A scheduling rule in place of FIFO, the QoS performance of routers can be largely enhanced in both heavy and light traffic conditions without using more sophisticated mechanisms, such as two separate queues to separate data packets with different classes of priority. The router model with WSPT-A provides a high level of QoS performance, in addition to providing performance stability. While WSPT-A could be easily implemented in a router, the drawback is a possibly high runtime overhead in the router under heavy traffic conditions because of more computation involved in WSPT-A than in FIFO. Hence, WSPT-A may be adopted by routers in networks that are not heavily loaded and can sacrifice some runtime overhead for better QoS. Additionally, we run simulation experiments under two traffic conditions. Since this study shows the promising results of using WSPT-A, further field tests of putting WSPT-A on routers under real Internet traffic conditions will help verify the results of this study and provide insights into the impact of the runtime overhead before employing WSPT-A on the Internet.

Acknowledgements

This work is sponsored by the Air Force Research Laboratory—Rome (AFRL-Rome) under grant number F30602-01-1-0510. The US government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of, AFRL-Rome, or the US Government.

References

[1] Gevros P, Crowcorft J, Kirstein P, Bhatti S. Congestion control mechanisms and the best effort service model. IEEE Network, May/June 2001;15(3):16–26.
[2] Huston G. Internet performance survival guide. New York: Wiley, 2000. p. 9.
[3] Braden R, Clark D, Shenker S. Integrated services in the Internet architecture: an overview, Request For Comments (Informational) 1633, Internet Engineering Task Force, June 1994. Available: http://www.ietf.org/rfc.html.
[4] Blake S, Black D, Carlson M, Davies E, Wang Z, Weiss W. An architecture for differentiated service, Request for Comments (Informational) 2475, Internet Engineering Task Force, Dec. 1998. Available: http://www.ietf.org/rfc.html.

[5] Kurose J. Open issues and challenges in providing Quality-of-Service guarantees in high-speed networks. ACM SIGCOMM Computer Communication Review 1993;23(1):6–15.
[6] Sabata B, Chatterjee S, Davis M, Sydir JJ, Lawrence TF. Taxonomy of QoS Specifications. In: Third Workshop on Object-Oriented Real-Time Dependable Systems, 1997. p. 100–7. Available: http://www.ietf.org/rfc.html.
[7] Nicols K, Jacobson V, Zhang L. A Two-bit Differentiated Services Architecture for the Internet, Request for Comments 2638, Internet Engineering Task Force, Nov. 1997.
[8] Parekh AK, Gallager RG. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. IEEE/ACM Transactions on Networking 1993;1(3):344–57.
[9] Pinedo M. Scheduling: theory, algorithms, and systems. Englewood Cliffs, NJ: Prentice-Hall; 1995. p. 27–32.
[10] Almquist P. Type of Service in the Internet Protocol Suite. Request for Comments 1349, Internet Engineering Task Force, July 1992. Available: http://www.ietf.org/rfc.html.
[11] Zhang H. Service disciplines for guaranteed performance service in packet-switching networks. In: Proceedings of the IEEE, October 1995. p. 1374–96.
[12] Stankovic J, Ramamritham K. Hard Real-Time Systems. Silver Spring, MD: IEEE Computer Society Press; 1988.
[13] Ng TSE, Stephens DC, Stoica I, Zhang H. Supporting BestEffort Traffic with Fair Service Curve. Proceedings of IEEE Globecom '99, Global Internet Symposium, December 1999.
[14] Bennett JCR, Benson K, Charny A, Courtney WF, Boudec JYL. Delay jitter bounds and packet scale rate guarantee for expedited forwarding. In: INFOCOM, 2001. p. 1502–9.
[15] Ferrari D. Delay Jitter Control Scheme for Packet-switching Internetworks. Computer Communications 1992;15(6):367–73. julho/agosto de 1992.
[16] Mansour Y, Patt-Shamir B. Jitter Control in QoS Networks, 39th Annual IEEE Symposium on Foundations of Computer Science. October 1998. p. 50–9.
[17] Zhang H. Providing end to end performance guarantees using non workconserving disciplines. Computer Communications 1995;18(10):769–81.
[18] Ye N. QoS-Centric stateful resource management in information systems. Information Systems Frontiers 2002;4(2):149–60.
[19] Lawrence TF. The quality of service model and high assurance. In: Proceedings of the IEEE High Assurance Systems Engineering Workshop, 1997.